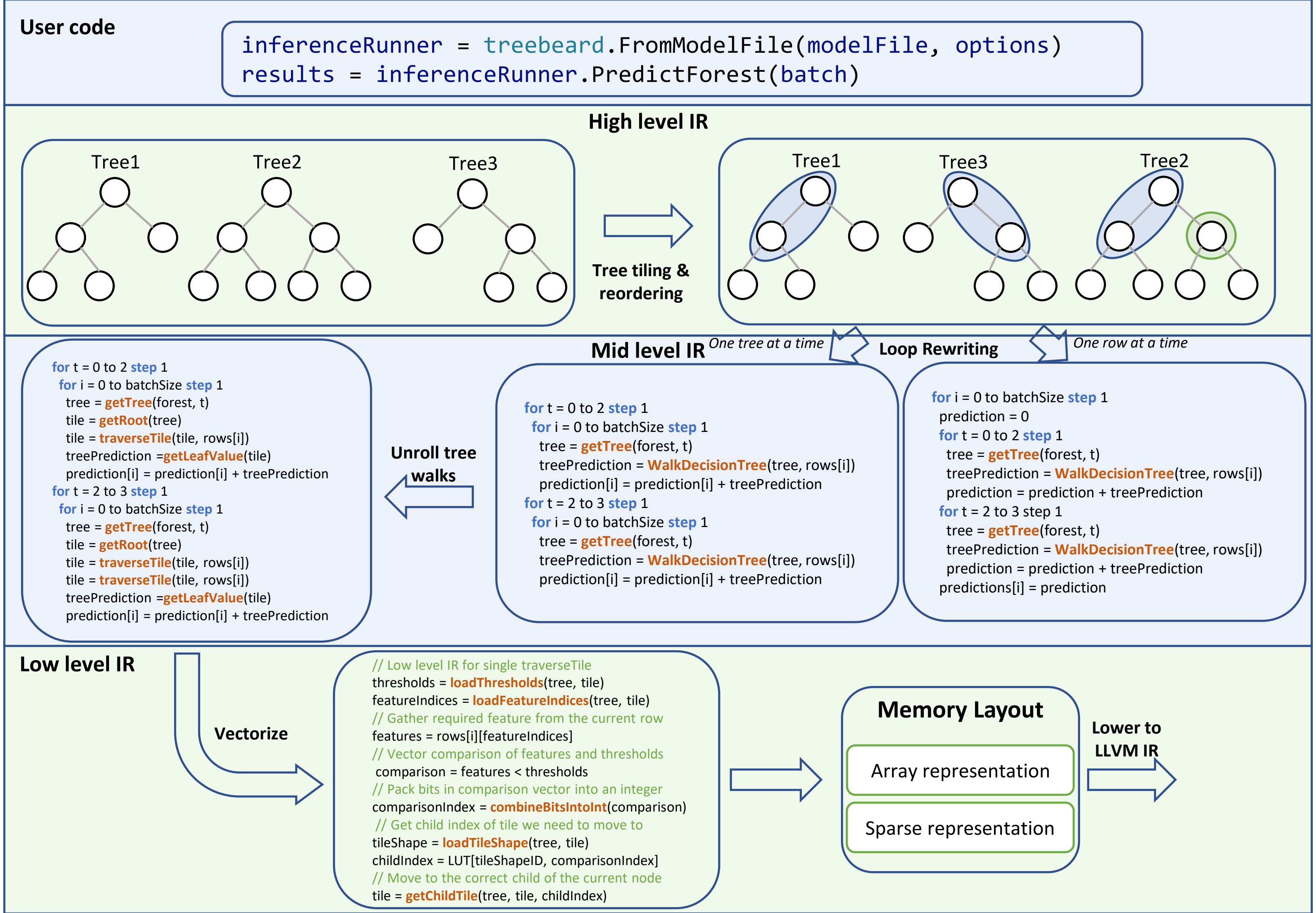

```
inferenceRunner = treebeard.FromModelFile(modelFile, options)
results = inferenceRunner.PredictForest(batch)
```

```
for t = 0 to 2 step 1
 for i = 0 to batchSize step 1
  tree = getTree(forest, t)
  tile = getRoot(tree)
  tile = traverseTile(tile, rows[i])
  treePrediction =getLeafValue(tile)
  prediction[i] = prediction[i] + treePrediction
for t = 2 to 3 step 1
 for i = 0 to batchSize step 1
  tree = getTree(forest, t)
  tile = getRoot(tree)
  tile = traverseTile(tile, rows[i])
  tile = traverseTile(tile, rows[i])
  treePrediction =getLeafValue(tile)
  prediction[i] = prediction[i] + treePrediction
```

```
for t = 0 to 2 step 1
 for i = 0 to batchSize step 1
  tree = getTree(forest, t)
  treePrediction = WalkDecisionTree(tree, rows[i])
  prediction[i] = prediction[i] + treePrediction
for t = 2 to 3 step 1
 for i = 0 to batchSize step 1
  tree = getTree(forest, t)
  treePrediction = WalkDecisionTree(tree, rows[i])
  prediction[i] = prediction[i] + treePrediction
```

```
for i = 0 to batchSize step 1
 prediction = 0
 for t = 0 to 2 step 1
  tree = getTree(forest, t)
  treePrediction = WalkDecisionTree(tree, rows[i])
  prediction = prediction + treePrediction
 for t = 2 to 3 step 1
  tree = getTree(forest, t)
  treePrediction = WalkDecisionTree(tree, rows[i])
  prediction = prediction + treePrediction
 predictions[i] = prediction
```

```
// Low level IR for single traverseTile
thresholds = loadThresholds(tree, tile)
featureIndices = loadFeatureIndices(tree, tile)
// Gather required feature from the current row
features = rows[i][featureIndices]
// Vector comparison of features and thresholds
 comparison = features < thresholds
// Pack bits in comparison vector into an integer
comparisonIndex = combineBitsIntoInt(comparison)
 // Get child index of tile we need to move to
tileShape = loadTileShape(tree, tile)
childIndex = LUT[tileShapeID, comparisonIndex]
// Move to the correct child of the current node
tile = getChildTile(tree, tile, childIndex)
```